

Code Club

2.26.21

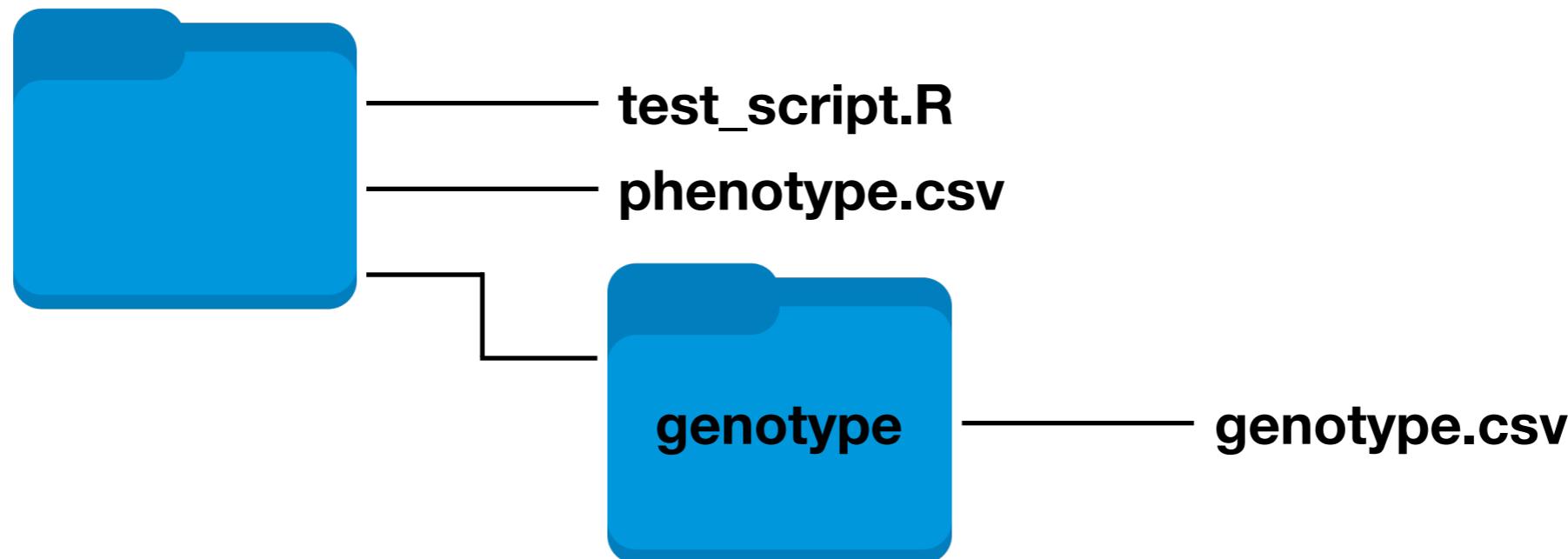


Format

- **Code hacking** – if someone is stuck on how to solve a tricky problem, we could discuss it. Alternatively, if someone WAS stuck and DID solve the problem... how did they do it?
- **Code review** – let's learn from each other! Maybe this involves sharing a few of your favorite functions/packages/tools you wrote. Maybe we work more formally with a “code swap”
- **Code workshops** – more formal presentation centered on a specific topic. Think Dan’s conda environment workshop.
- **Choose your own adventure** – We aren’t limited to any specific format! This is for YOUR benefit. This is an informal, judgement-free space to share ideas and learn from each other!

My
FAVORITE
THINGS

Working directory



```
1 # set working directory as the location of the active file!
2 setwd(glue::glue("{dirname(rstudioapi::getActiveDocumentContext()$path)}/"))
3
4 # read in data
5 data <- read.csv("phenotype.csv")
6 data2 <- read.csv("genotype/genotype.csv")
7
8
```

Thanks, Tim!

glue::glue()

```
library(tidyverse)

# with paste
new_starwars <- starwars %>%
  dplyr::mutate(character = paste(name, ", (", homeworld, ")"), sep = ""))

# with paste0
new_starwars <- starwars %>%
  dplyr::mutate(character = paste0(name, ", (", homeworld, ")"))

# with glue
new_starwars <- starwars %>%
  dplyr::mutate(character = glue::glue("{name}, ({homeworld})"))
```

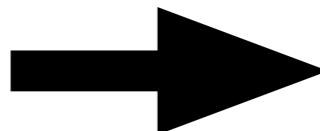
glue::glue() increases readability, but requires namespace...

tidyverse::separate_rows()

```
library(tidyverse)
data(starwars)

# separate each film to new line
new_starwars <- starwars %>%
  dplyr::select(name, films) %>%
  tidyverse::separate_rows(films, sep = ",")
```

name	films
Luke Skywalker	c("Revenge of the Sith", "Return of the Jedi", "The Empire Strikes Back")
C-3PO	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith")
R2-D2	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith")
Darth Vader	c("Revenge of the Sith", "Return of the Jedi", "The Empire Strikes Back")
Leia Organa	c("Revenge of the Sith", "Return of the Jedi", "The Empire Strikes Back")
Owen Lars	c("Attack of the Clones", "Revenge of the Sith", "A New Hope")
Beru Whitesun Lars	c("Attack of the Clones", "Revenge of the Sith", "A New Hope")
R5-D4	A New Hope
Biggs Darklighter	A New Hope
Obi-Wan Kenobi	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith")
Anakin Skywalker	c("Attack of the Clones", "The Phantom Menace", "Revenge of the Sith")
Wilhuff Tarkin	c("Revenge of the Sith", "A New Hope")



name	films
Luke Skywalker	c("Revenge of the Sith")
Luke Skywalker	"Return of the Jedi"
Luke Skywalker	"The Empire Strikes Back"
Luke Skywalker	"A New Hope"
Luke Skywalker	"The Force Awakens"
C-3PO	c("Attack of the Clones")
C-3PO	"The Phantom Menace"
C-3PO	"Revenge of the Sith"
C-3PO	"Return of the Jedi"
C-3PO	"The Empire Strikes Back"
C-3PO	"A New Hope")
R2-D2	c("Attack of the Clones")

to split one cell into several rows

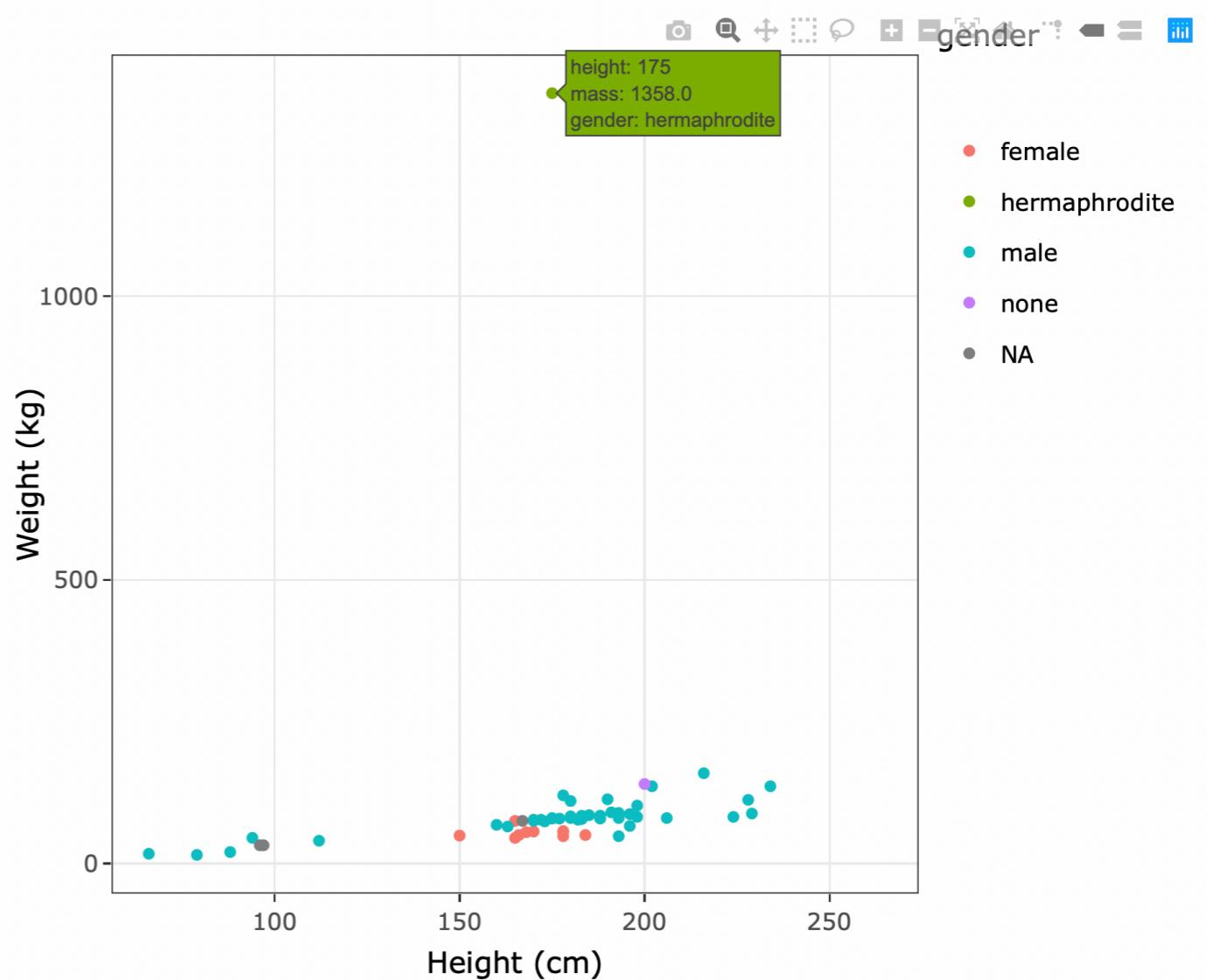
dplyr::case_when()

plotly::ggplotly()

```
library(tidyverse)
library(plotly)

# make normal ggplot
plot <- starwars %>%
  ggplot2::ggplot(.) +
  ggplot2::aes(x = height, y = mass, color = gender) +
  ggplot2::geom_point() +
  ggplot2::theme_bw(14) +
  ggplot2::labs(x = "Height (cm)", y = "Weight (kg)")

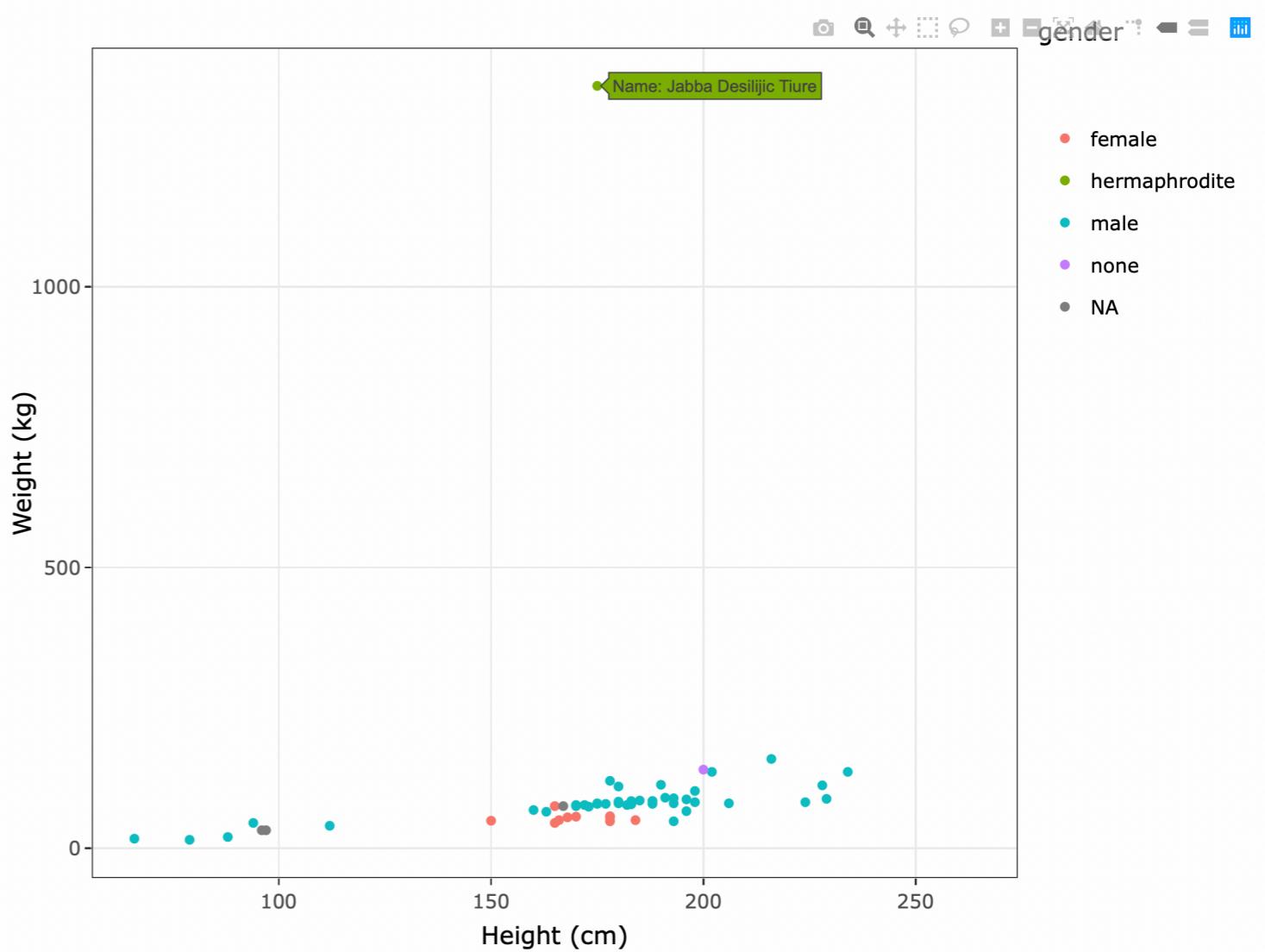
# make plotly
plotly::ggplotly(plot)
```



plotly::ggplotly()

```
# add custom text with pointer
plot <- starwars %>%
  ggplot2::ggplot(.) +
  ggplot2::aes(x = height, y = mass, color = gender,
    text = glue::glue("Name: {name}")) +
  ggplot2::geom_point() +
  ggplot2::theme_bw(14) +
  ggplot2::labs(x = "Height (cm)", y = "Weight (kg)")

plotly::ggplotly(plot, tooltip = "text")
```



DT::datatable()

DT::datatable(new_starwars)

	name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species	eye_color_new
1	Luke Skywalker	172	77	blond	fair	blue	19	male	Tatooine	Human	cool
2	C-3PO	167	75		gold	yellow	112		Tatooine	Droid	warm
3	R2-D2	96	32		white, blue	red	33		Naboo	Droid	warm
4	Darth Vader	202	136	none	white	yellow	41.9	male	Tatooine	Human	warm
5	Leia Organa	150	49	brown	light	brown	19	female	Alderaan	Human	dark
6	Owen Lars	178	120	brown, grey	light	blue	52	male	Tatooine	Human	cool
7	Beru Whitesun lars	165	75	brown	light	blue	47	female	Tatooine	Human	cool
8	R5-D4	97	32		white, red	red			Tatooine	Droid	warm
9	Biggs Darklighter	183	84	black	light	brown	24	male	Tatooine	Human	dark
10	Obi-Wan Kenobi	182	77	auburn, white	fair	blue-gray	57	male	Stewjon	Human	cool

DT::datatable(new_starwars, filter = "top")

	name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species	eye_color_new
All	All	All	All	All	All	All	All	All	All	All	All
1	Luke Skywalker	172	77	blond	fair	blue	19	male	Tatooine	Human	cool
2	C-3PO	167	75		gold	yellow	112		Tatooine	Droid	warm
3	R2-D2	96	32		white, blue	red	33		Naboo	Droid	warm
4	Darth Vader	202	136	none	white	yellow	41.9	male	Tatooine	Human	warm

gganimate()

```
# create plot with animation
myplot <- traitmap %>%
  dplyr::mutate(rownum = as.numeric(row.names(.))) %>%
  ggplot2::ggplot(.) +
  ggplot2::aes(x = pos/1e6, y = maxlod) +
  ggplot2::geom_line() +
  ggplot2::theme_bw(12) +
  ggplot2::labs(x = "Genomic position", y = "LOD") +
  ggplot2::facet_grid(~chr, scales = "free_x", space = "free_x") +
  ggplot2::theme(panel.grid = ggplot2::element_blank(),
                legend.position = "none") +
  # transition_reveal used to draw lines over "time" aka rownum
  gganimate::transition_reveal(rownum)

# animate with specific dimensions
gganimate::animate(myplot, height = 3, width = 8, units = "in", res = 150)

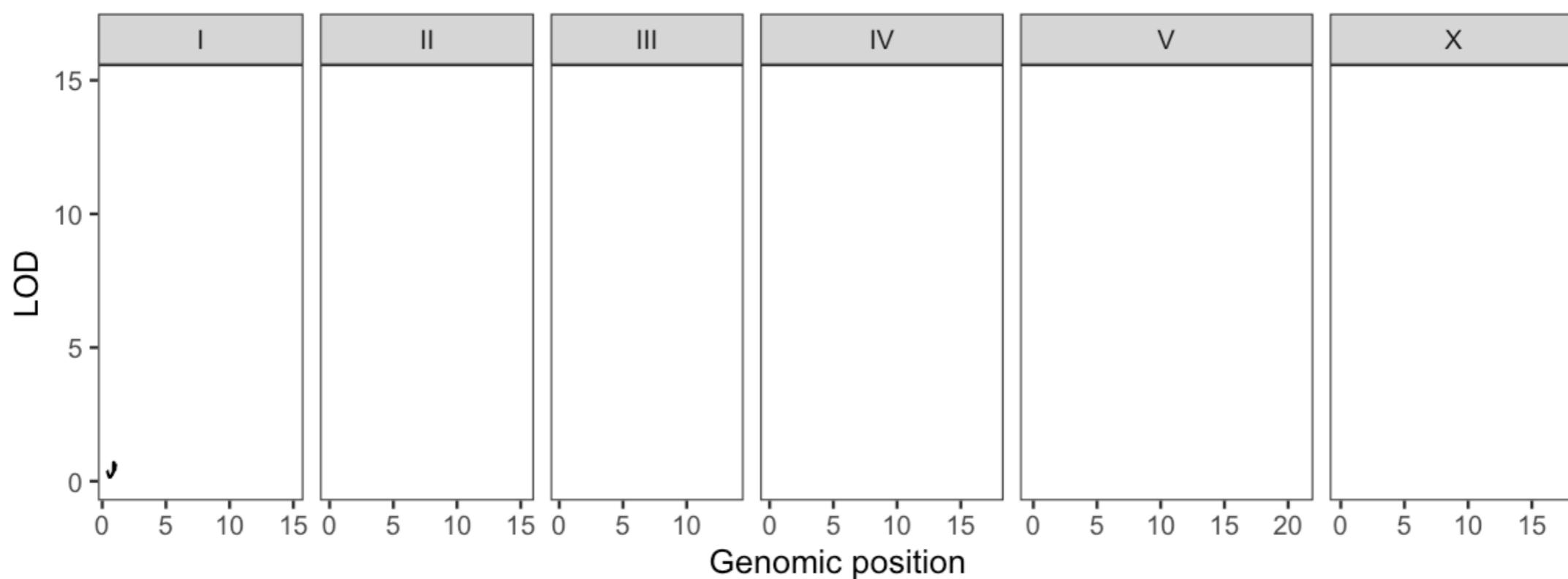
# save
gganimate::anim_save("lodplot_animate.gif")
```

gganimate()

```
# create plot with animation
myplot <- traitmap %>%
  dplyr::mutate(rownum = as.numeric(row.names(.))) %>%
  ggplot2::ggplot(.) +
  ggplot2::aes(x = pos/1e6, y = maxlod) +
  ggplot2::geom_line() +
  ggplot2::theme_bw(12) +
  ggplot2::labs(x = "Genomic position", y = "LOD") +
  ggplot2::facet_grid(~chr, scales = "free_x", space = "free_x") +
  ggplot2::theme(panel.grid = ggplot2::element_blank(),
                legend.position = "none") +
  # transition_reveal used to draw lines over "time" aka rownum
  gganimate::transition_reveal(rownum)

# animate with specific dimensions
gganimate::animate(myplot, height = 3, width = 8, units = "in", res = 150)

# save
gganimate::anim_save("lodplot_animate.gif")
```



% <> %

Allows you to make changes to a current data frame
and save over that dataframe

NOT part of tidyverse, need to load magrittr separately

```
library(tidyverse)
library(magrittr)

# with magrittr
new_starwars %<>% dplyr::filter(eye_color_new == "cool")

# with normal tidyverse convention
new_starwars <- new_starwars %>%
  dplyr::filter(eye_color_new == "cool")
```

* not sure how I feel about this one yet *

data.table::fread()

What is your favorite way to read in .tsv and .csv data??

read.table()

readr::read_tsv()

read.delim()

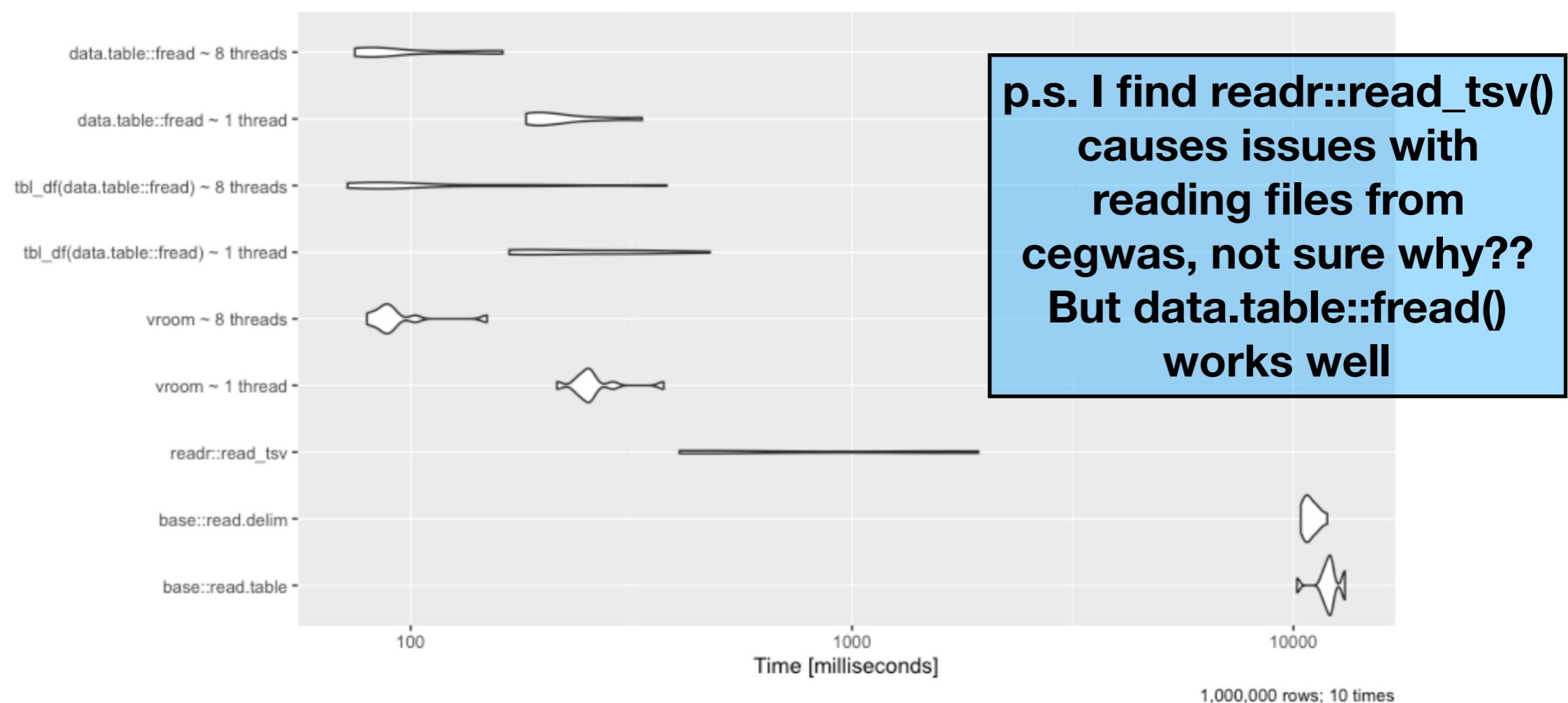
data.table::fread()

read.csv()

vroom::vroom()

other??

`data.table::fread()`



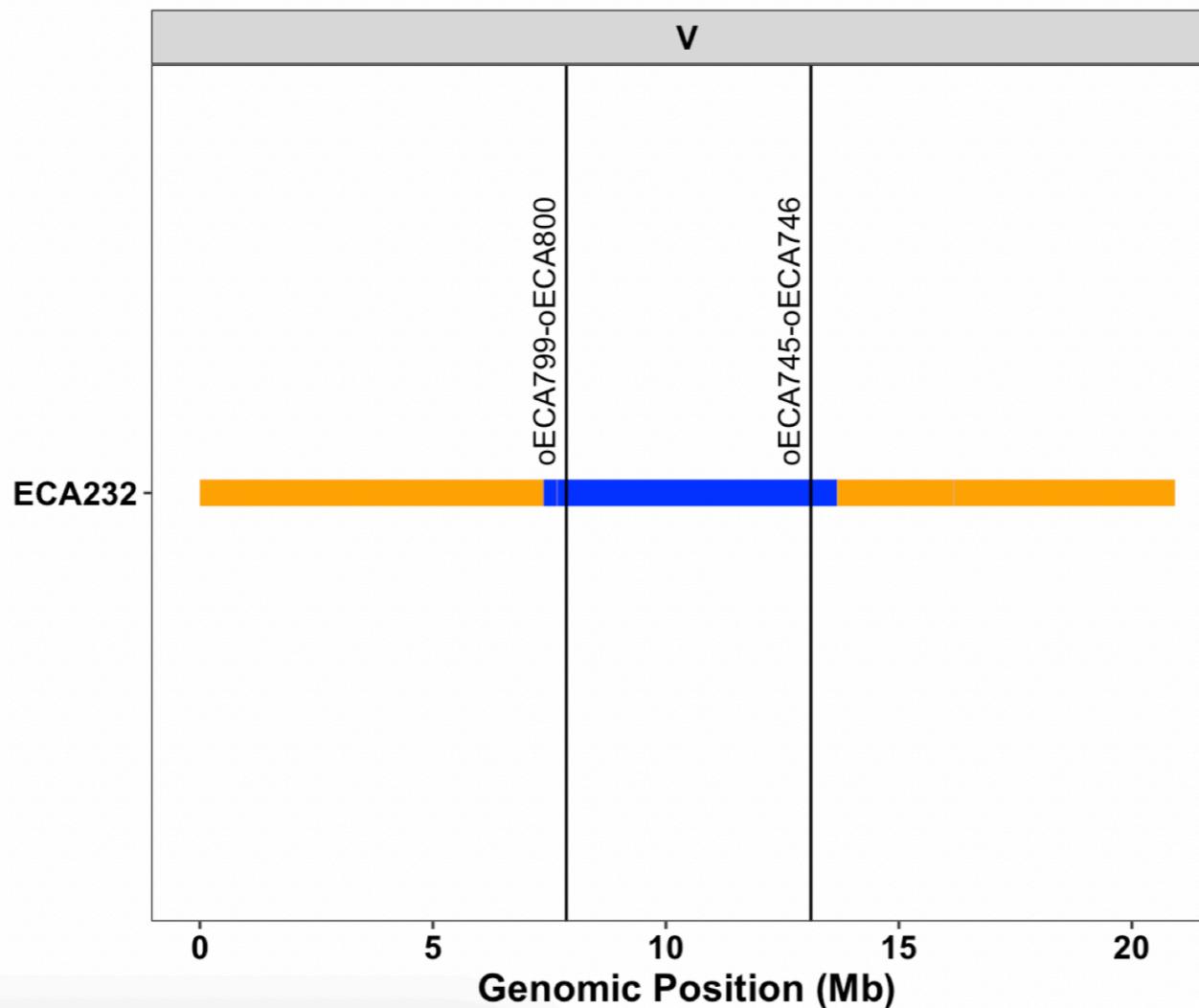
Looks like the base R functions lose - by a lot. `data.table::fread` and `vroom::vroom` come out on top at ~ 100 millesconds whereas the base functions take ~10 seconds or 100x longer!

Stop wasting your time with `read.table`, `read.csv`, and `read.delim` and move to something quicker like `data.table::fread`, or `vroom::vroom` both of which perform much faster. Both can also take advantage of multiple cores but outperform base R even when they only use a single thread!

Plot primer positions

```
source("~/Dropbox/AndersenLab/LabFolders/Katie/scripts_kse/NIL_narrowing.R")

# plot primers
plot_primers(c("oECA799-oECA800",
               "oECA745-oECA746"), NIL = "ECA232", chr = "V")
```



```

# Function to plot NIL genotype and primer locations
# primer pairs - pair of primers in the format "oECAXXX-oECAXXX", can be multiple
# NIL - ECAXXX strain name to plot genotype (N2 is default)
# RIL = false. If true, will plot RILs instead of NILs
plot_primers <- function(primer_pairs, NIL = "N2", RIL = F, chr = c("I", "II", "III", "IV", "V", "X")) {

  # get primer positions
  primer_positions <- primers %>%
    dplyr::filter(pair %in% primer_pairs | pair2 %in% primer_pairs)

  if(RIL == T) {
    df <- rilgeno
  } else {
    df <- nilgeno
  }

  # plot NIL genotype
  df %>%
    dplyr::filter(sample %in% NIL) %>%
    dplyr::filter(chrom %in% chr) %>%
    dplyr::mutate(gt = as.character(gt)) %>%
    ggplot2::ggplot(.) +
    ggplot2::geom_segment(ggplot2::aes(x = start/1e6, y = sample, xend = end/1e6, yend = sample, color = gt, size = 2)) +
    ggplot2::facet_grid(~chrom, scales = "free", space = "free") +
    ggplot2::scale_color_manual(values=c("1"="orange", "2"="blue")) +
    ggplot2::theme_bw() +
    ggplot2::theme(axis.text.x = ggplot2::element_text(size=12, face="bold", color="black"),
                  axis.text.y = ggplot2::element_text(size=12, face="bold", color="black"),
                  axis.title.x = ggplot2::element_text(size=14, face="bold", color="black"),
                  axis.title.y = ggplot2::element_text(size=14, face="bold", color="black"),
                  strip.text = ggplot2::element_text(size = 12, face = "bold", color = "black"),
                  plot.title = ggplot2::element_text(size=24, face="bold"),
                  legend.position = "none",
                  panel.grid.minor = ggplot2::element_blank(),
                  panel.grid.major = ggplot2::element_blank()) +
    ggplot2::labs(x = "Genomic Position (Mb)", y = "") +
    ggplot2::geom_vline(data = data.frame(pos = primer_positions$pos, chrom = primer_positions$Chromosome),
                        ggplot2::aes(xintercept = pos)) +
    ggplot2::geom_text(data = data.frame(pos = primer_positions$pos,
                                         chrom = primer_positions$Chromosome,
                                         strain = NIL[length(NIL)],
                                         primers = primer_positions$pair), aes(x = pos, y = strain, label = primers),
                       angle = 90,
                       hjust = -0.1,
                       vjust = -0.5)
}

```

**p.s. I just added the `ggplot2::`
right before this meeting... 😂**

Housekeeping



How often to meet?

Monthly? Twice monthly?



Volunteers?

Message Katie!

Code Club: Suggestion Box

Please fill out HERE and add to at any time!



Code Club - Suggestion Box

Share your ideas for what you would like to learn about or discuss during Andersen Lab Code Clubs!

AndersenLab: Coding Best Practices

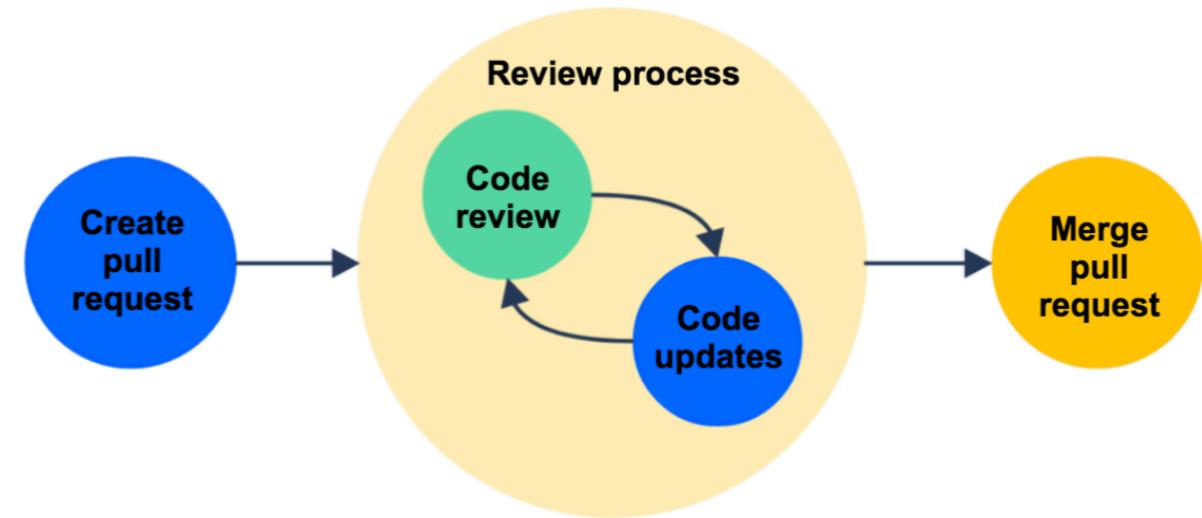
Take a look HERE and add suggestions!

AndersenLab: Coding Best Practices

General

- You should be doing most (if not all) of your analyses in AndersenLab/LabFolders/Dropbox/YourName (except for Quest -- see below)
 - This is (1) to make sure the data is backed up/saved with version history and (2) to allow other lab members to access your code/scripts when necessary
- Do NOT use spaces in names of files or folders. Try to not use spaces in column names too (although sometimes it is necessary for a final table output).
 - Computers often have a hard time reading spaces and code used to ignore spaces can vary from program to program.
 - Instead, you can use "_" or "." or "-" or capitalization (fileName.txt)
- NEVER replace raw data!!!!!!

Code Review?



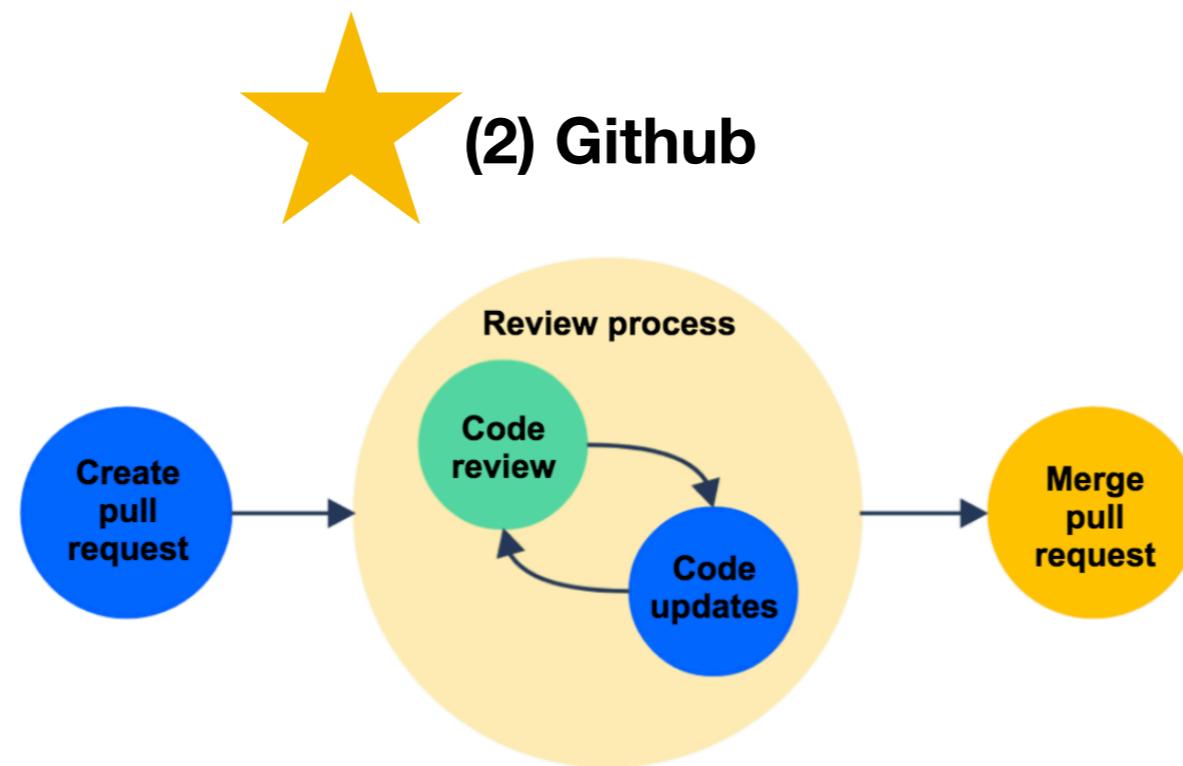
Code Review

(1) Simple

- Katie sends file to Tim
- Tim adds comments and new updates
- Tim sends file back to Katie
- Katie reviews Tim's comments/suggestions

(3) Other

??



Homework in between meetings: one code review and one code reviewed?